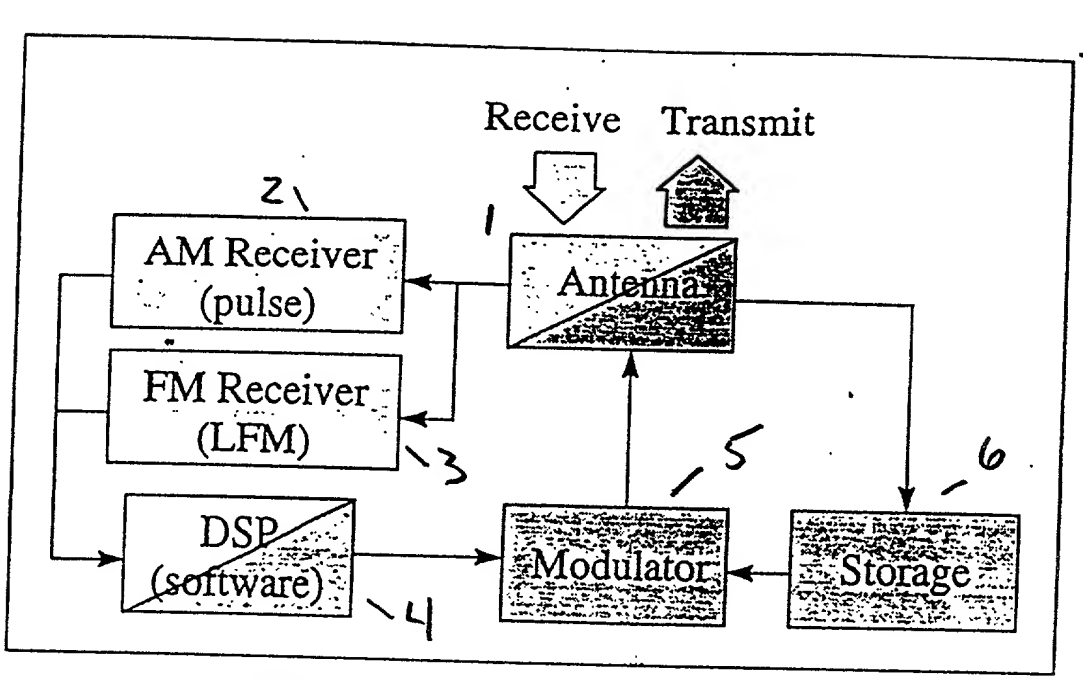
10
Receive
Transmit
2
1
AM Receiver
(pulse)
Antenna
FM Receiver
(LFM)
3
5
6
DSP
(software)
Modulator
Storage
4

Figure 1

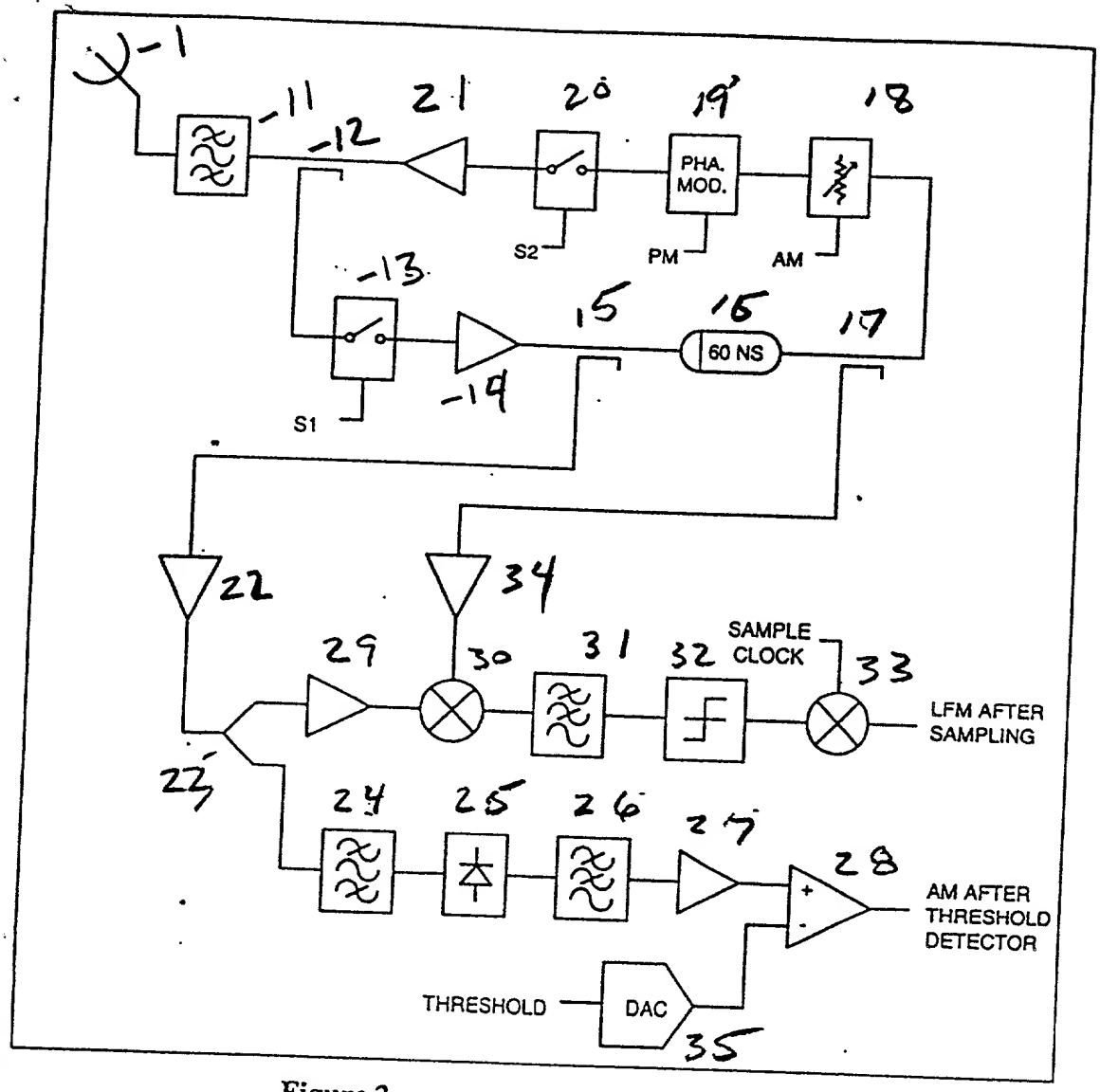
PHA. MOD.
PM
AM
S2
S1
60 NS
SAMPLE CLOCK
LFM AFTER SAMPLING
AM AFTER THRESHOLD DETECTOR
THRESHOLD
DAC

Figure 2

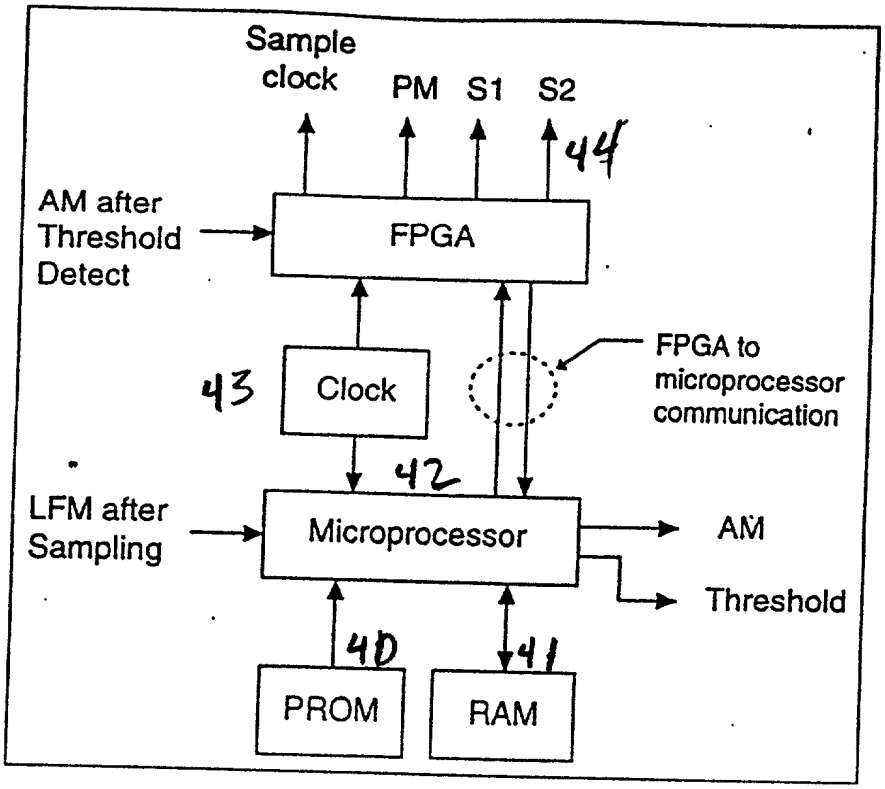
Sample clock
PM
S1
S2
44
AM after Threshold Detect
FPGA
43
Clock
FPGA to microprocessor communication
42
LFM after Sampling
Microprocessor
AM
Threshold
40
41
PROM
RAM

Figure 3

4

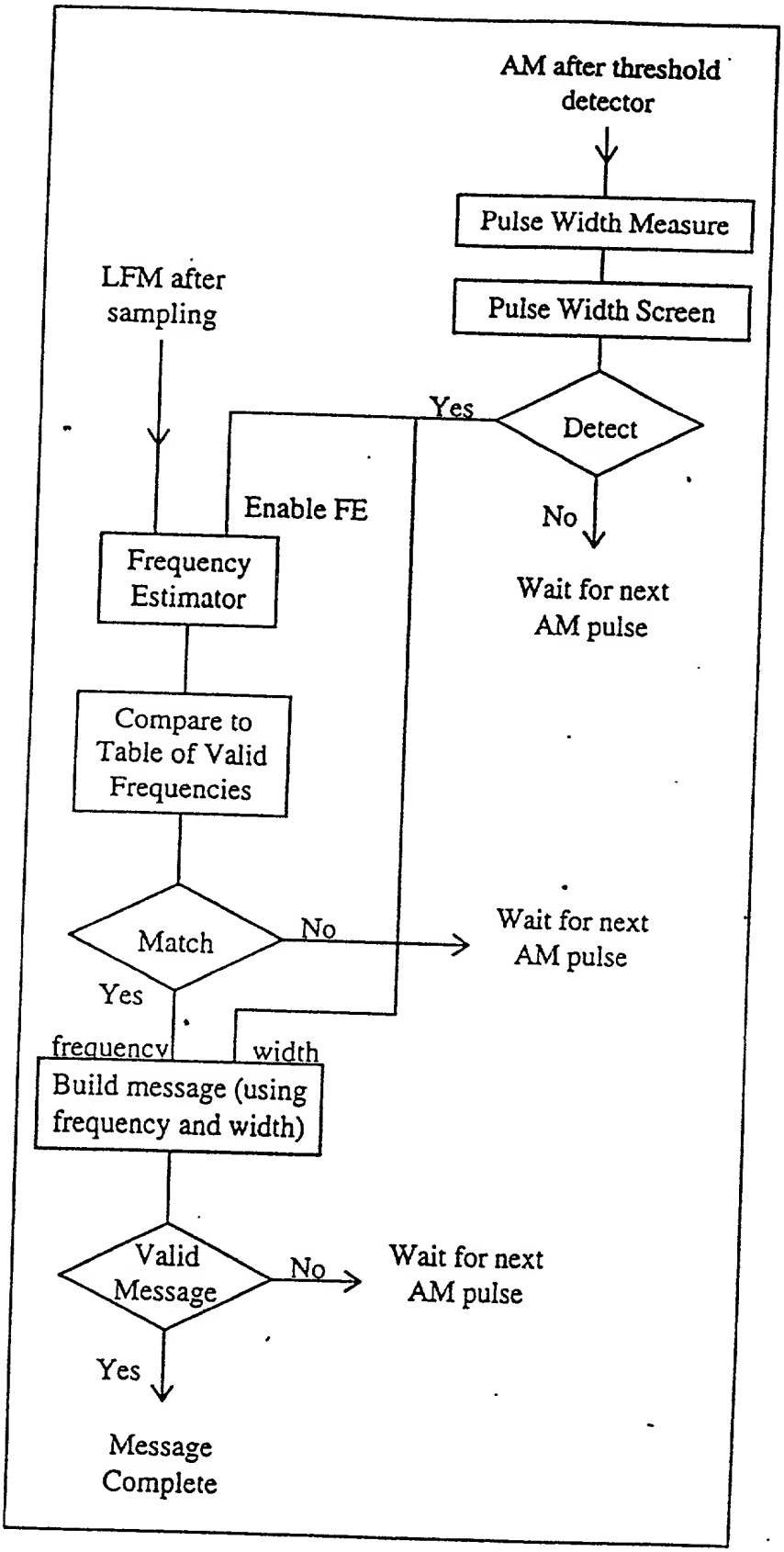
AM after threshold detector
Pulse Width Measure
Pulse Width Screen
LFM after sampling
Yes
Detect
No
Enable FE
Frequency Estimator
Wait for next AM pulse
Compare to Table of Valid Frequencies
Match
No
Wait for next AM pulse
Yes
frequency
width
Build message (using frequency and width)
Valid Message
No
Wait for next AM pulse
Yes
Message Complete

Figure 4

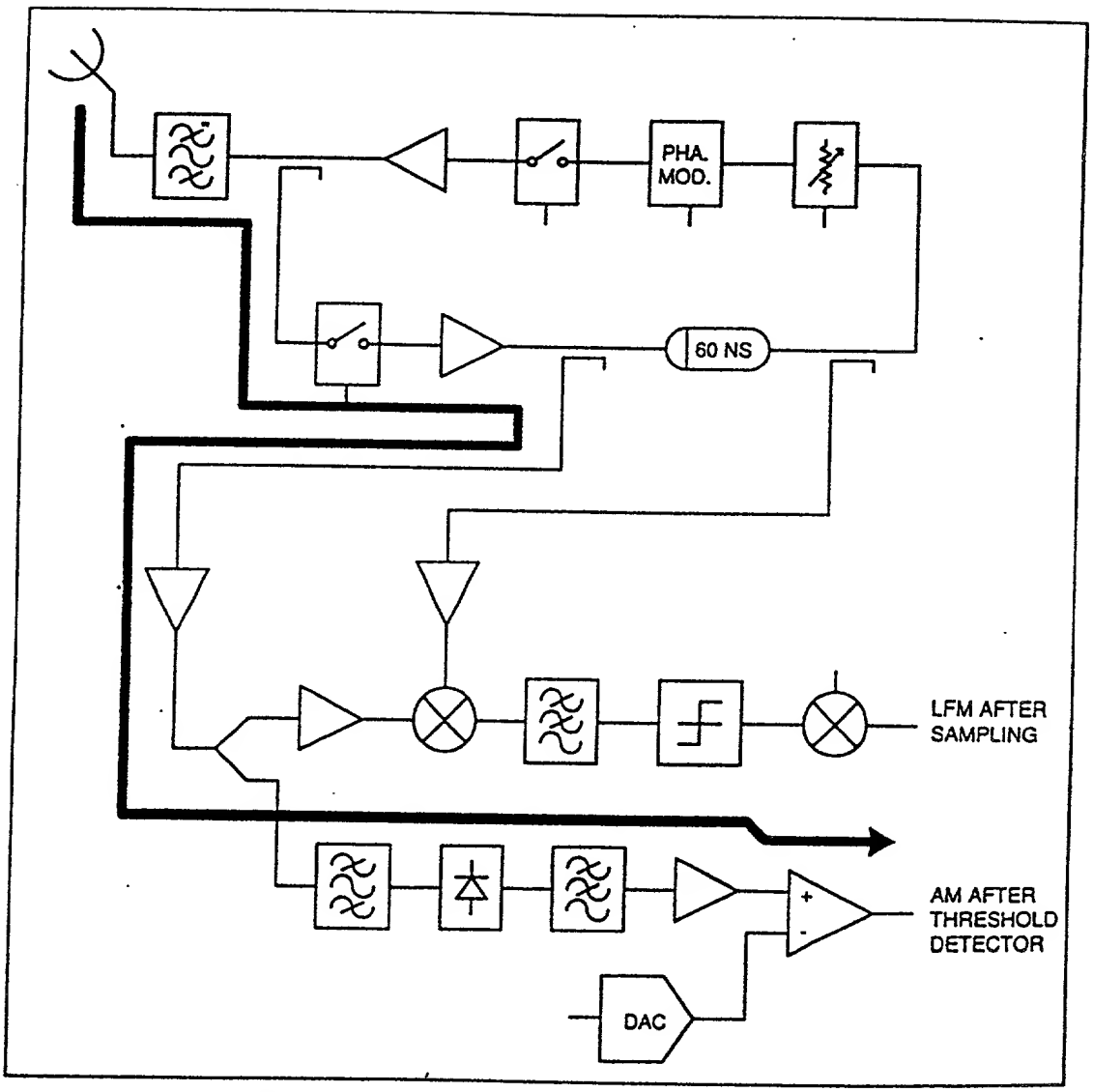
PHA.
MOD.
60 NS
LFM AFTER
SAMPLING
AM AFTER
THRESHOLD
DETECTOR
DAC

Figure 5

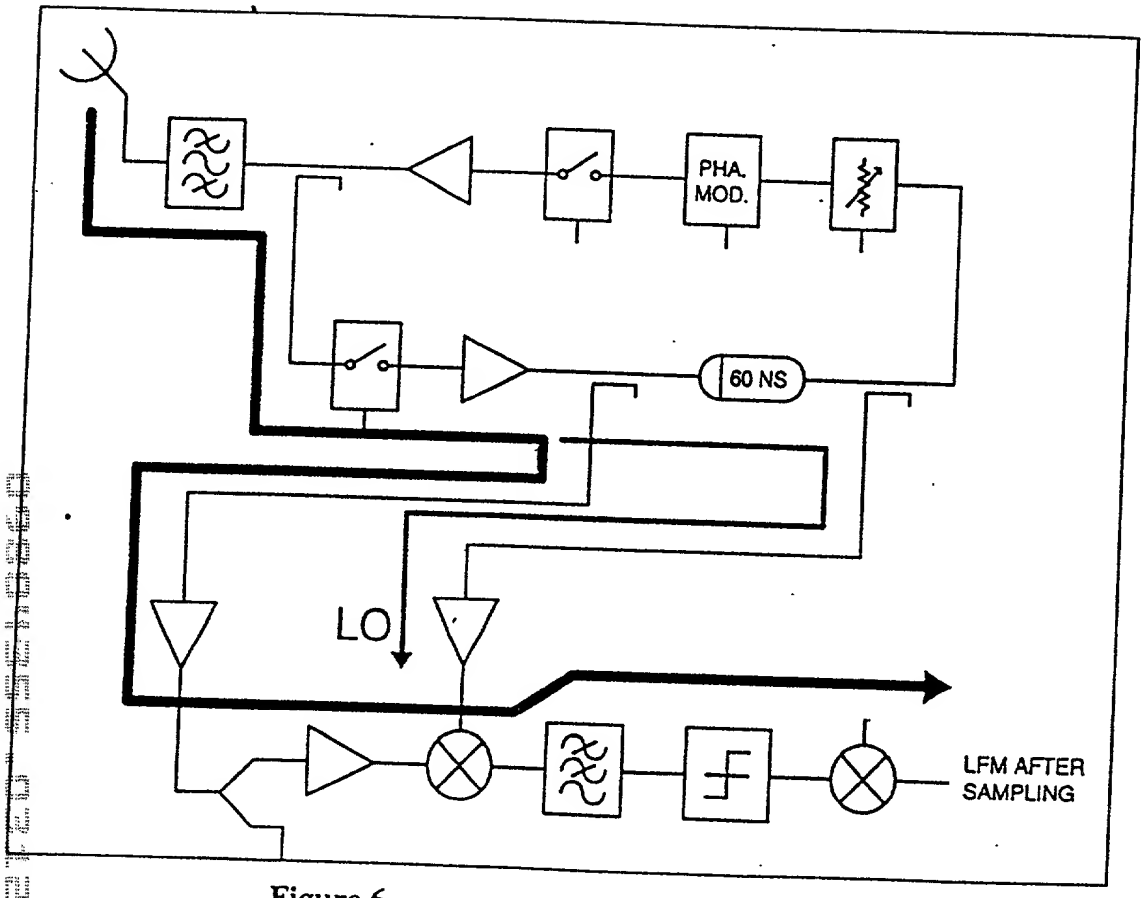

Figure 6

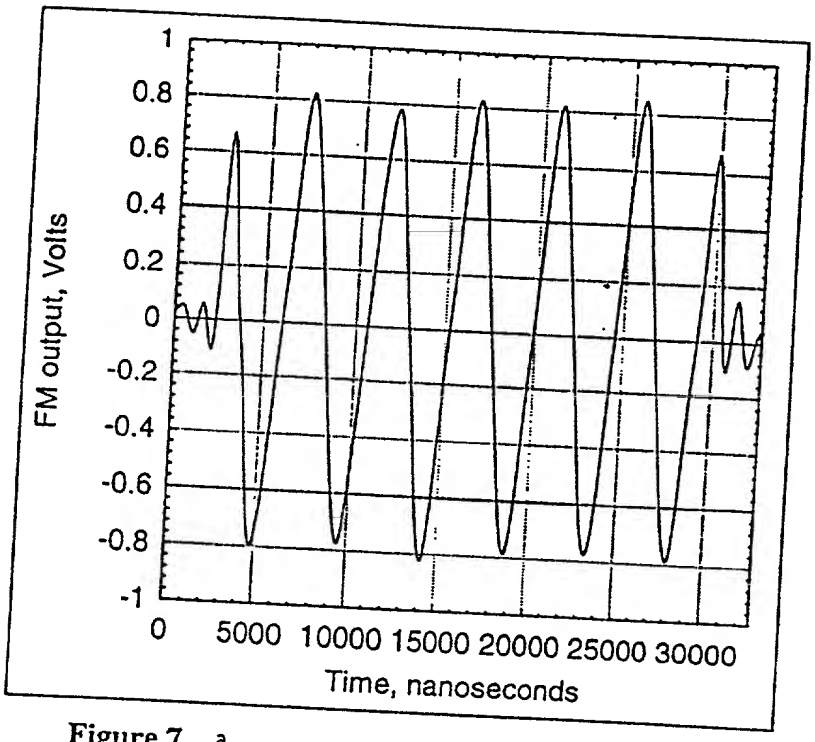
FM output, Volts
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5000
10000
15000
20000
25000
30000
Time, nanoseconds

Figure 7 a

```
transition = 0
cycles = 0
cycles_temp = 0
clock = 2e6
for (n = 2 ; n <= data ; n++)
{
      if ((SS[n] == 1) && (SS[n-1] == 1))
      {
            cycles_temp = cycles_temp + 1
            cycles = cycles + 1
            if (SLIN[n-1] != SLIN[n])
            {
                  transition = transition + 1
                  cycles_temp = 1
            } else {
            }
            if (transition == 0)
            {
                  cycles = 0
                  cycles_temp = 0
            } else {
            }
      } else {
      }
}
cycles = cycles - cycles_temp
if (cycles <= 0)
{
      cycles = 1
} else {
}
frequency_estimate = (transition - 1) * clock / (cycles * 2)
```

Figure 7b

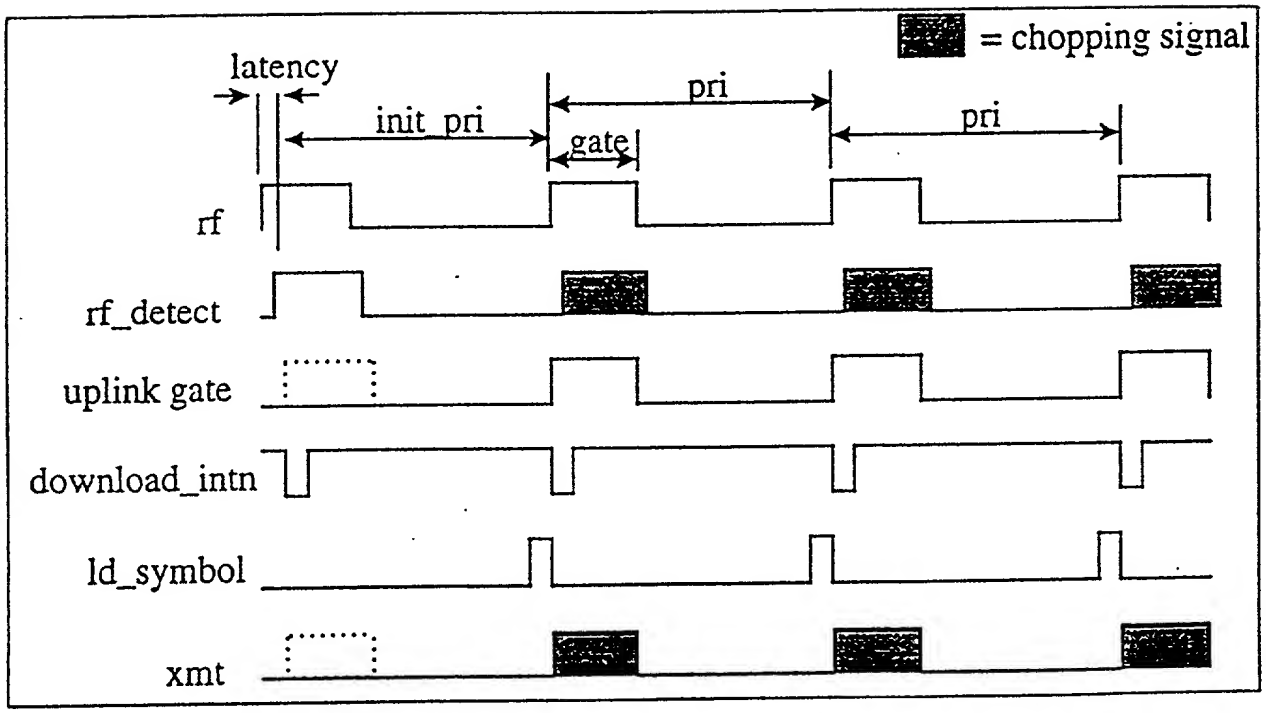
= chopping signal
latency
init_pri
pri
gate
pri
rf
rf_detect
uplink gate
download_intn
ld_symbol
xmt

Figure 8

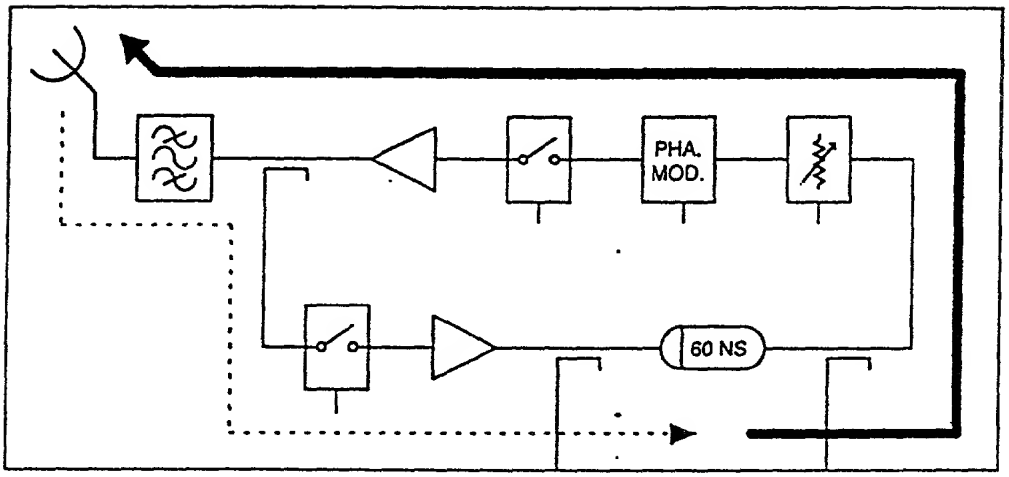

Figure 9

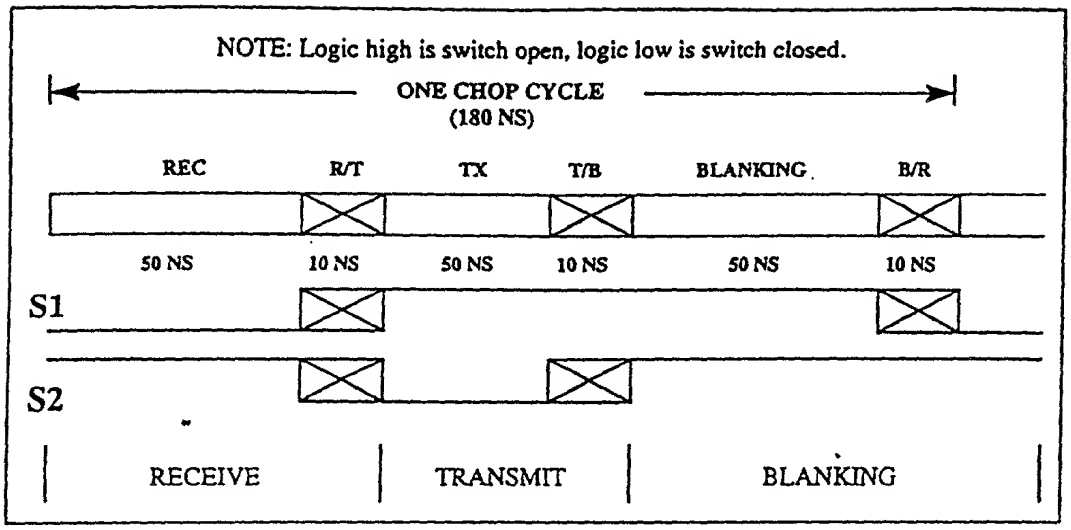
NOTE: Logic high is switch open, logic low is switch closed.
ONE CHOP CYCLE
(180 NS)
REC
R/T
TX
T/B
BLANKING
B/R
50 NS
10 NS
50 NS
10 NS
50 NS
10 NS
S1
S2
RECEIVE
TRANSMIT
BLANKING

Figure 10

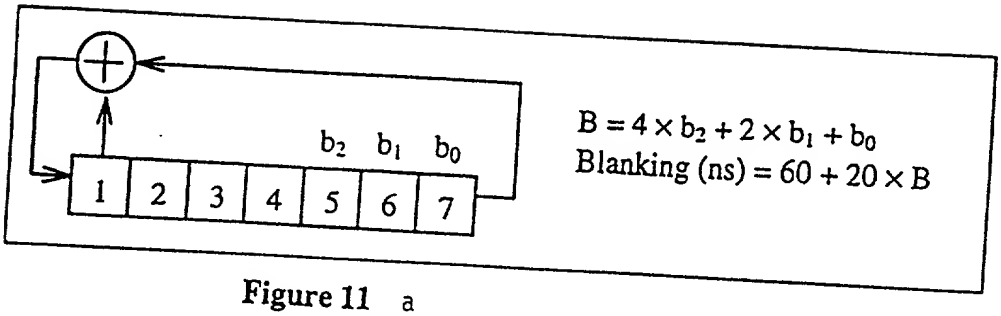

Figure 11 a

DATA BIT	AVERAGE FREQUENCY DEVIATION (Hertz)	PHASE (degrees)
000	+3100586	+90
001	+1025391	+90
010	-3100586	+90
011	-1025391	+90
100	+3100586	-90
101	+1025391	-90
110	-3100586	-90
111	-1025391	-90

Figure 11b

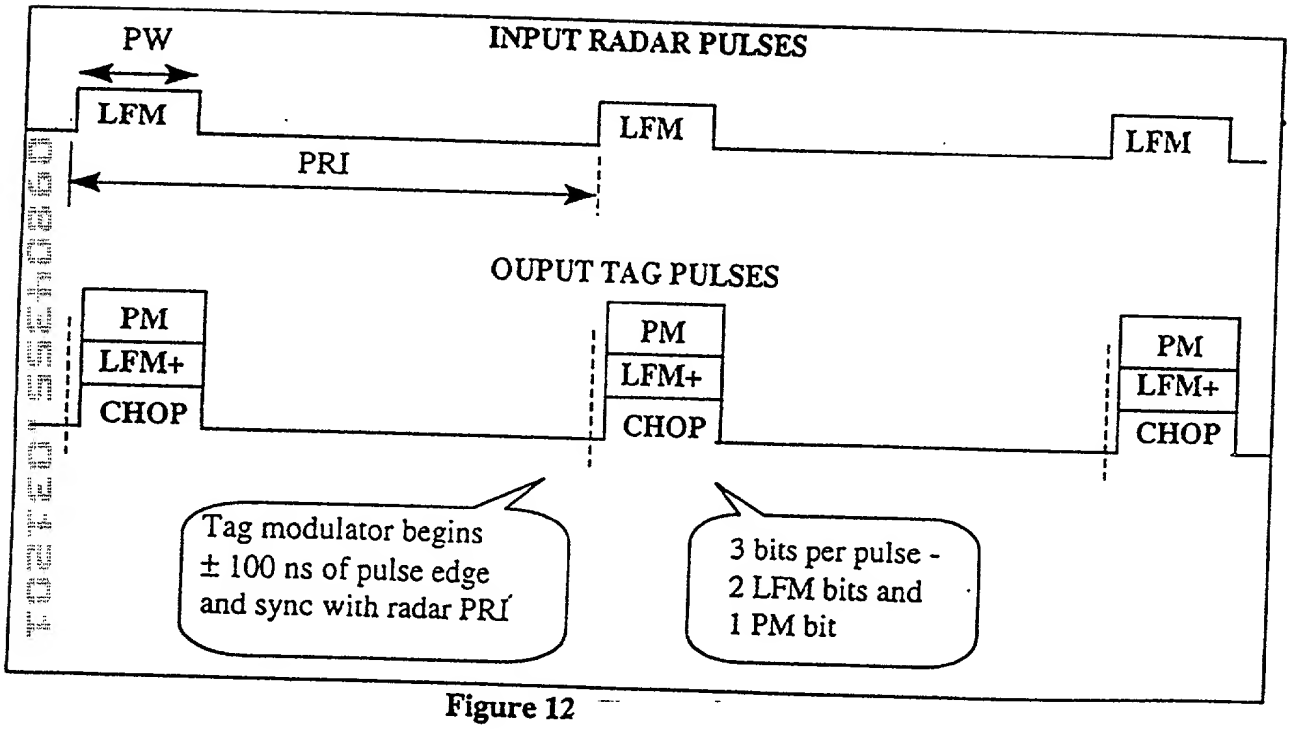
PW
INPUT RADAR PULSES
LFM
LFM
LFM
PRI
OUPUT TAG PULSES
PM
LFM+
CHOP
PM
LFM+
CHOP
PM
LFM+
CHOP
Tag modulator begins
± 100 ns of pulse edge
and sync with radar PRI
3 bits per pulse -
2 LFM bits and
1 PM bit

Figure 12

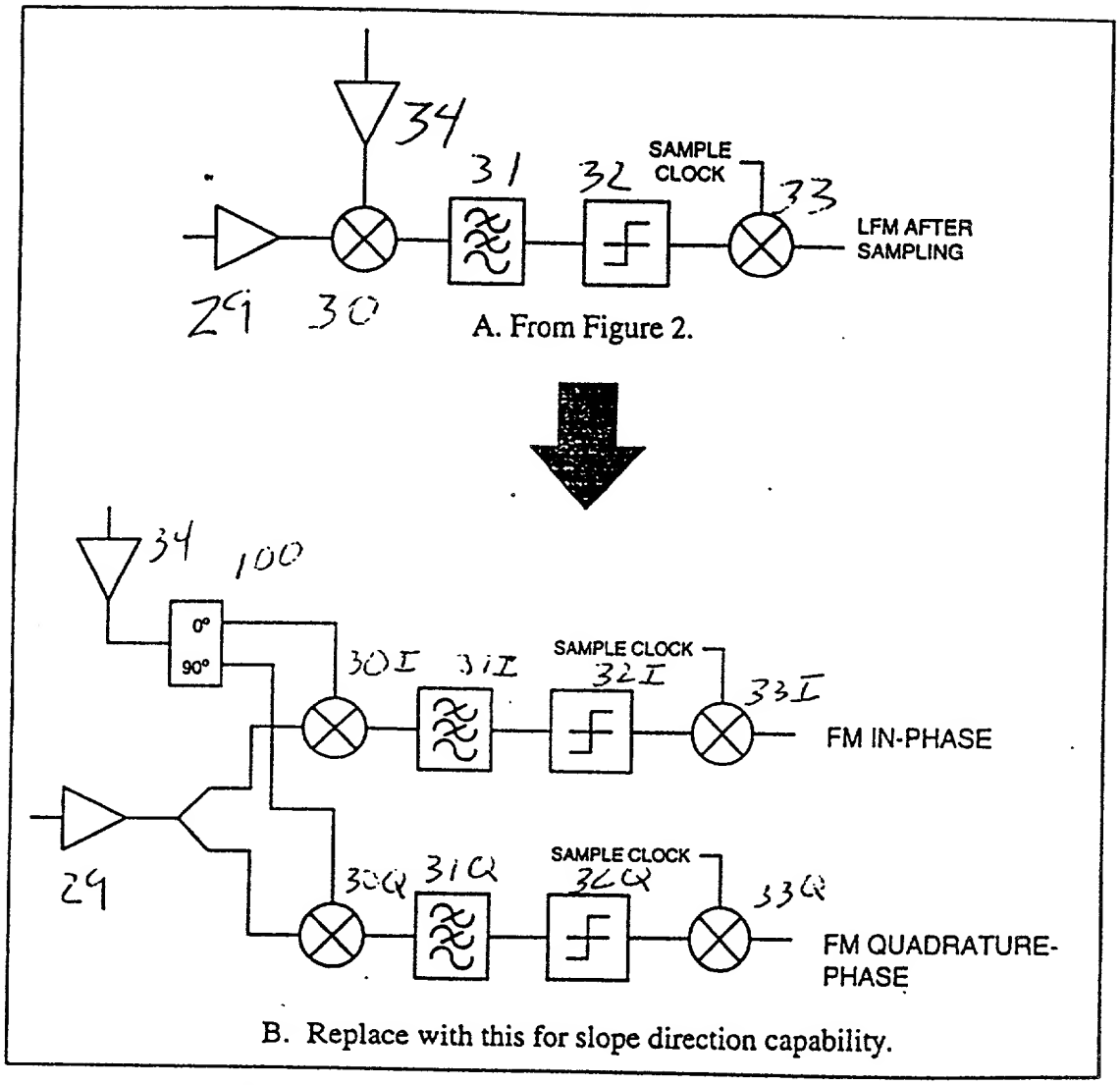
34
31
32
SAMPLE CLOCK
33
LFM AFTER SAMPLING
29
30
A. From Figure 2.
34
100
0°
90°
30I
31I
SAMPLE CLOCK
32I
33I
FM IN-PHASE
29
30Q
31Q
SAMPLE CLOCK
32Q
33Q
FM QUADRATURE-PHASE
B. Replace with this for slope direction capability.

Figure 13

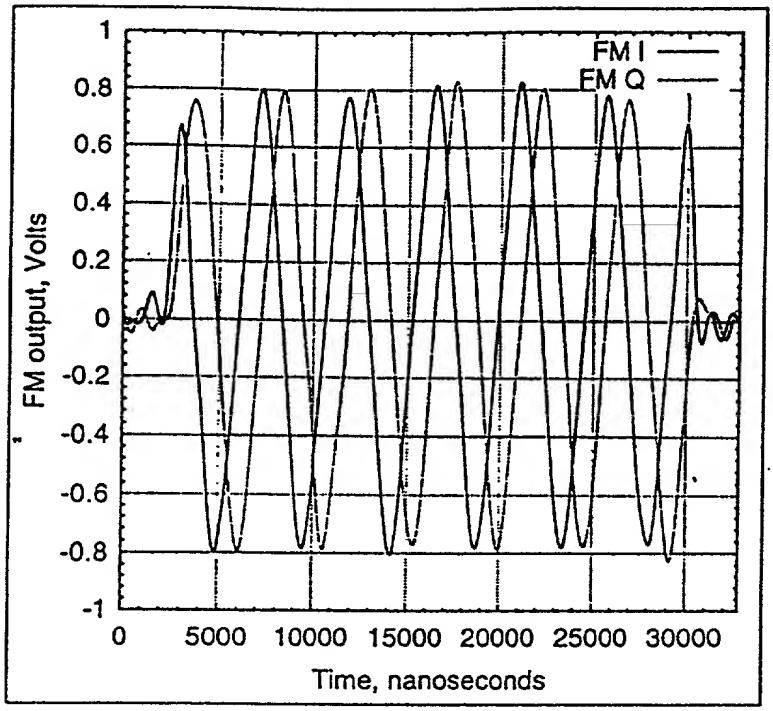

Figure 14a

```
for (n = 2 ; n <= data ; n++)
{
      if ((SS[n] == 1) && (SS[n-1] == 1))
      {
            if ((SLIN_Q[n-1] != SLIN_Q[n]) && SLIN_Q[n] == 1))
            {
                  SLOPE[n] = SLIN_I[n]
            }
      }
}
```

Figure 14b

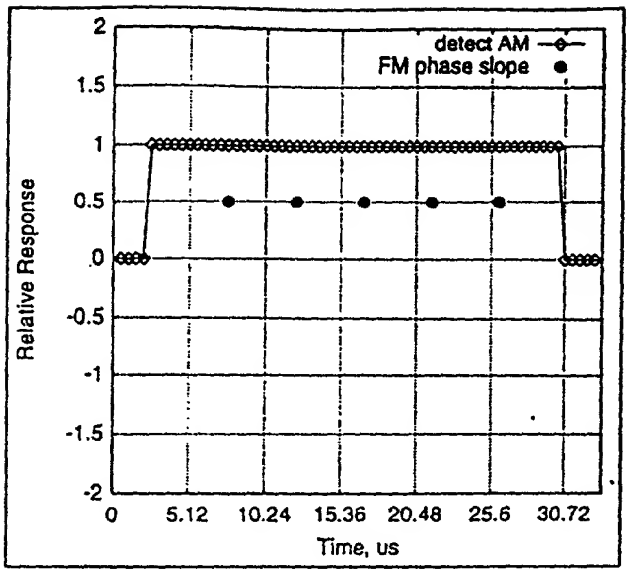

Figure 15

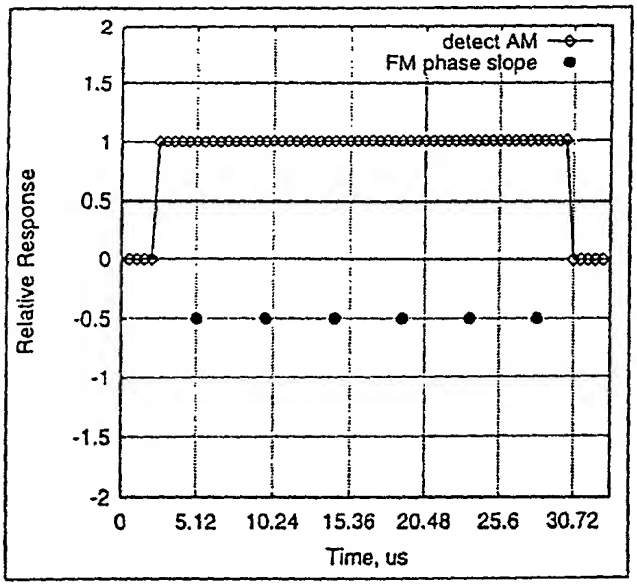

Figure 16